Reference trajectory modification based on spatial iterative learning for contour control of 2-axis NC systems

Article (Accepted Version)

Reference Trajectory Modification based on Spatial Iterative Learning for Contour Control of 2-axis NC Systems

Jiangang Li, *Member, IEEE,* Yiming Wang, Yanan Li*, *Member, IEEE,* and Wenshu Luo

***Abstract*—Contour error is a main factor that affects the quality of products in numerical control (NC) machining. This paper presents a contour control strategy based on digital curves for high-precision control of computer numerical control (CNC) machines. A contour error estimation algorithm is presented for digital curves based on a geometrical method. The dynamic model of the motion control system is transformed from time domain to space domain because the contour error is dependent on space instead of time. Spatial iterative learning control (sILC) is developed to reduce the contour error, by modifying the reference trajectory in the form of G code. This allows system improvement without interference of low-level controllers so it is applicable to many commercial controllers where interpolators and feed-drive controllers cannot be altered. The effectiveness of this method is verified by experiments on a NC machine, which have shown good performance not only for smooth trajectories but also for large curvature trajectories.**

***Index Terms*—Contour error control, iterative learning control, CNC machines.**

I. Introduction

In CNC machining, the machining path is a result of coordinated multi-axis motion and the machining quality of a product is usually evaluated by the contour error (the geometric deviation of the actual contour from the desired one). Therefore, reducing the contour error is an important task of multi-axis motion control systems [1], [2].

In order to improve the accuracy of contour tracking in CNC machining, a lot of research effort is devoted to feed-forward/feedback control strategies to improve the accuracy of single-axis tracking, thereby indirectly improving the accuracy of motion control systems. For example, zero phase error tracking controller proposed in [3] achieved tracking control by eliminating the zero poles in the system transfer function. However, the main cause of the contour error is the dynamic mismatch between the axes, so simply improving the tracking accuracy of a single axis does not necessarily lead to a reduction in the contour error [4], [5]. In order to solve this problem for multi-axis motion systems, cross-coupling control (CCC) was proposed in [6], [7] where the coupling effects of each axis on others was studied. Researchers have since then followed this pioneering work and improved its performance by focusing on two major issues: contour error estimation (CEE) and contour error control (CEC) [8], [9].

CEE is a trivial task for a linear trajectory, as one can compute the real contour error. However, a trajectory in CNC machining is usually a free curve, for which various methods were developed to estimate the contour error. A free curve contour error estimation model based on tangent was proposed in [10]. A local approximation method was proposed in [11], which has a large error when the tracking error is large. In [12], a contour error estimation method was proposed based on position backtracking, which is in the form of small line segments. However, the outputs of a CNC system are usually digital curves, namely a series of discrete data points, so extensive curve-based analysis is infeasible. Moreover, it is not reasonable to estimate the contour error of the curve by linear interpolation. In [13], a contour error prediction and compensation method was developed for five-axis systems.

A lot of effort has been also made for CEC. In [14], the contour error was reduced by modifying feedback in combination with integral sliding mode control. In [15], the traditional CCC scheme was improved for CEC. In [16], position-loop feedforward control was studied to reduce the contour error. In [17], a multi-input multi-output linear parameter-varying feedback controller was proposed for contour error minimization in CNC machines. In [18], a discrete-time fractional-order sliding-mode CEC was proposed to improve the contour tracking accuracy. However, these methods at the control level are not easy to implement, especially for commercial controllers which are not open to users. Moreover, contour control at the control level may have adverse effects on the controller and may change the original speed planning. In [19], [20], trajectory planning was modified to achieve reduction of the contour error, but the modification of the trajectory planning is limited by many constraints. In [21], an error compensation approach of parametric programming was developed that deals with the deformation of the machining system caused by radial cutting force in turning, and the generated error was compensated to the tool path. This approach works well with many commercial controllers for which the interpolators and the feed-drive controllers are not allowed to be altered, and it is simple to implement. In order to further improve the contour tracking of CCC, especially in mass production, the strategy of cross iteration learning coordination control (CCILC) was developed, which combines iterative learning

This research is supported by National Key R&D Program of China under grant number 2019YFB1703700 and National Natural Science Foundation of China under grant number U1913213.

J. Li, Y. Wang and W. Luo are or were with the Harbin Institute of Technology Shenzhen, China 518055.

Y. Wang is also with the 28th Research Institute of China Electronics Technology Group Corporation, Nanjing, China 210007.

*Correspondence: Y. Li is with the Department of Engineering and Design, University of Sussex, Brighton BN1 9RH, UK. Email: yl557@sussex.ac.uk

control (ILC) and CCC [22]. A novel algorithm that integrates ILC with empirical mode decomposition (EMD) was proposed to improve learning process [23].

As acknowledged by some of aforementioned works, ILC is a suitable method to eliminate the effects caused by repetitive interference in the CNC machining. It works in an iterative manner to gradually counteract the effects of repetitive disturbances. In [24], ILC was used to update control signals based on path errors and the time elapsed along a trajectory. While the ILC used in most of the existing works was based on a time period, the contour error is naturally defined in space. Therefore, any delay in one axis will introduce an error to the contour error compensation. In the field of ILC, spatial iterative learning control (sILC) was introduced for systems with spatial periodicity. In [25], [26], the standard temporal ILC algorithm was formulated with spatial variables and a sILC framework was developed based on 2D spatial convolution. In [27], sILC was developed for a class of high-order nonlinear motion systems in the presence of both parametric and nonparametric uncertainties, but it was also implemented in the control level.

Based on above discussions, this paper will introduce a new method to compute the contour error and to modify the reference trajectory without modification of the controller. We will use Hermite spline interpolation to get a cubic spline that is close to a free curve and then develop a geometrical method to estimate its contour error. This method does not require complex computation and is easy to implement, while it ensures a high estimation accuracy even for large curvature trajectories. We will develop sILC with modification of reference trajectory achieved by modifying G code. This method allows system improvement without interference of the low-level controllers. This property is favourable as for many commercial controllers the interpolators and feed-drive controllers are not allowed to be altered. We will show that this learning guarantees convergence of the contour error when the iteration number is large enough, by using a composite energy function. Finally, we will present experimental results to verify the validity of our idea.

In the rest of the paper, we first introduce an improved contour error estimation method in Section II. In Section III, we propose a contour control method for modifying the reference trajectory based on sILC. The convergence of contour error is proved in theory. In Section IV, we present the results of the experiment under different conditions for comparison. At last, we draw conclusions of this paper and discuss possible future works.

II. Contour error estimation

In this section, the approach of contour error estimation for digital curves will be described. First, the Hermite spline interpolation method will be used to deal with a digital curve, so that a better approximation of the free curve can be achieved and mathematical analysis could be carried out. Then, the contour error is estimated based on closest point search and geometrical analysis.

A. Digital curve processing

The original reference trajectory used to calculate the contour error is obtained by discretization of a parametric curve, and the actual trajectory is obtained by machine tool sampling. Therefore, both of them are digital curves.

In order to estimate the contour error more accurately, splines are used to construct these curves. In particular, cubic splines are used as they make a reasonable compromise between the flexibility and the computing speed [28]. According to different methods to determine the coefficient, cubic splines have three forms: cubic natural splines, Hermite splines and Cardinal splines. In this work, the Hermite spline method is selected as any part of the curve only depends on the local control point.

A general Hermite spline can be described as

$$P(t) = A_i l^3 + B_i l^2 + C_i l + D_i \qquad (l \in [0,1]) \tag{1}$$

where $P(l) = [x(l)\ y(l)\ z(l)]$ represents position vector with $x(l), y(l), z(l)$ as coordinates and l as spline parameter. A_i, B_i, C_i and D_i are coefficients to be determined in the following.

Suppose the starting point and ending point of a curve segment are respectively $P_0,\ P_1$, and the tangent vectors of the curve segment at the two points are respectively $P_0^{'},\ P_1^{'}$. Then, the following equations can be obtained that provide solutions of the four coefficients in Eq. (1):

$$\begin{cases} P_0 = A_i l^3 + B_i l^2 + C_i l + D_i = D_i (l=0) \\ P_1 = A_i l^3 + B_i l^2 + C_i l + D_i = A_i + B_i + C_i + D_i (l=1) \\ P_0^{'} = C_i + 2B_i l + 3A_i l^2 = C_i (l=0) \\ P_1^{'} = C_i + 2B_i l + 3A_i l^2 = C_i + 2B_i + 3A_i (l=1) \end{cases} \tag{2}$$

Let us write Eq. (2) in a form of matrices, as below

$$\begin{bmatrix} P_0 \\ P_1 \\ P_0^{'} \\ P_1^{'} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} D_i \\ C_i \\ B_i \\ A_i \end{bmatrix} \tag{3}$$

Then, with solved A_i, B_i, C_i and D_i, we can obtain the matrix expression for the Hermite spline curve as below

$$P(t) = \begin{bmatrix} l^3 & l^2 & l & 1 \end{bmatrix} \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ P_0^{'} \\ P_1^{'} \end{bmatrix} \tag{4}$$

Finally, we can obtain the expression of the k-th segment curve as

$$\begin{aligned} P(l) = {} & P_k(2l^3 - 3l^2 + 1) + P_{k+1}(-2l^3 + 3l^2) \\ & + P_k^{'}(l^3 - 2l^2 + l) + P_{k+1}^{'}(l^3 - l^2) \end{aligned} \tag{5}$$

B. Estimation algorithm

The estimation of contour error is based on digital curve processing in the previous subsection, followed by searching the closest point and Frenet coordinate. The detailed process is given as follows.

Consider a trajectory Γ shown in Fig. 1, where $Q(k)$ represents the actual tool position at time k, $P(k)$ represents

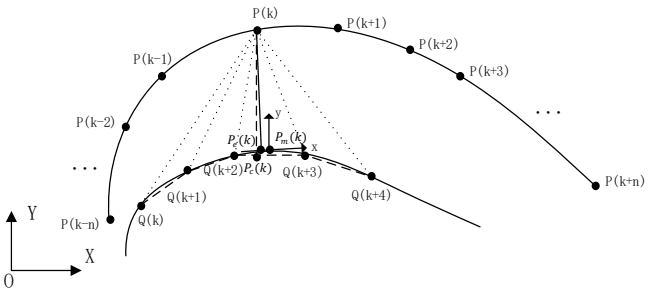


Fig. 1: Computation of contour error

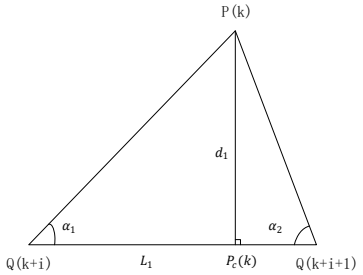


Fig. 2: Computation of pedal point

the tool instruction position and $Q(k) = (x_{qk}, y_{qk})$, $P(k) = (x_{pk}, y_{pk})$.

In a process of repetitive iterations, $P(k)$ is invariant, so the contour error is defined based on the deviation from $P(k)$. The idea is to compute the contour error at the same position in different iterations, so that we can implement contour control at the right place. When we compute the contour error at the position $P(k)$, the first step is to find the actual position point closest to $P(k)$. Then, we can use the following equation to compute the distance between $P(k)$ and actual position, and determines the point $Q(k+n)$ that is closest to $P(k)$:

$$L = \sqrt{(x_{pm} - x_{qk})^2 + (y_{pm} - y_{qk})^2} \tag{6}$$

Note that the distance range of search is within that between the last pedal point $P_c(k-1)$ and the ending point. Therefore, there are two cases to find the closest point. When $P_c(k-1)$ is the closest point, we have one segment L_1 to compute the pedal point. When $Q(k+n)$ is the closest point, we have two segments L_1 and L_2 to compute the pedal point. In the next step, we use the closest point to compute the new pedal point based on trigonometry as shown in Fig. 2. The calculation process is divided into the following three cases:

(1) If $\alpha_1 < 90°$ or $\alpha_2 < 90°$, the pedal point is on the segment. We can compute $P_c(k)$ by the following two equations in segment L_1:

$$\begin{cases} |P(k)P_c(k)| = |P(k)Q(k+n)|sin\alpha_1 \\ |Q(k+n)P_c(k)| = |P(k)Q(k+n)|cos\alpha_1 \end{cases} \tag{7}$$

$$P_c(k) = Q(k+n) + \frac{|Q(k+n)P_c(k)|}{|Q(k+n)Q(k+n+1)|}\overrightarrow{Q(k+n)Q(k+n+1)} \tag{8}$$

(2) If $\alpha_1 = 90°$ or $\alpha_2 = 90°$, we can get $Q(k+n)$ or $Q(k+n+1)$ as the pedal point $P_c(k)$.

(3) If $\alpha_1 > 90°$ or $\alpha_2 > 90°$, there is no pedal point in segment L_1, so we need to compute the pedal point in L_2; if there is no pedal point in L_2, the point $Q(k+n)$ can be regarded as the contour error point.

Based on the method of curve interpolation, we can get the equation of a curve between two points as below

$$\begin{cases} x_i(l) = A_{i1}l^3 + B_{i1}l^2 + C_{i1}l + D_{i1} & l \in [0,1] \\ y_i(l) = A_{i2}l^3 + B_{i2}l^2 + C_{i2}l + D_{i2} & l \in [0,1] \end{cases} \tag{9}$$

Moreover, we compute the spline parameter l as below

$$l = \frac{\left|\overrightarrow{Q(k+n)P_c(k)}\right|}{\left|\overrightarrow{Q(k+n)Q(k+n+1)}\right|} \tag{10}$$

From Eqs. (9) and (10), we can get the coordinate $P_m(k) = (x_{mk}, y_{mk})$ in the spline curve, and then a coordinate system is established with this point as the origin, the tangent as X-axis and normal as Y-axis, and the contour error as the Y-coordinate.

From Eq. (9), we can get the tangent vector $\vec{\tau}$ at the point $P_m(k)$, then the contour error $e = \|P_m(k)Q(k)\|\vec{\tau}$. At the same time, we can get the contour error point $P_e(k)$.

C. Verification of contour error estimation

In order to verify the correctness of our contour error estimation method, we use a circular trajectory with radius of $1.5mm$ to conduct an experiment. The real contour error of the circular trajectory can be computed as follows:

$$e_r = |\vec{x}| - r \tag{11}$$

where $\vec{x}$ is the actual position vector with the center of circle as the origin and r is the radius of circle.

The results with the above analytic method and our computation method are shown in Fig. 3 (a), where we can find that the contour errors computed by two methods almost overlap. The difference between them is shown in Fig. 3 (b), with the maximum of the difference as 5.9×10^{-5}mm, much smaller than the contour error itself. These results indicate that our contour error computation method achieves accurate estimation.

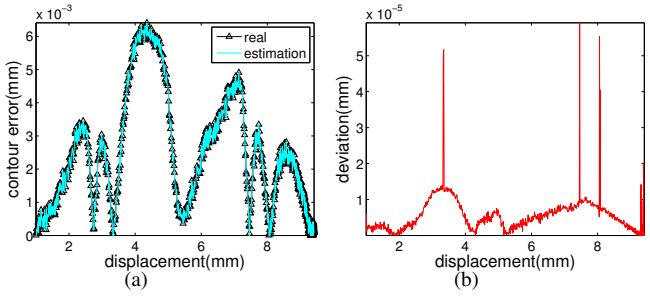


Fig. 3: Contour error (a) and difference between contour errors computed by the analytic method and the proposed one (b)

III. Dynamics analysis and contour error control

In this section, the control structure is established, the dynamics model of a CNC system is analyzed and the approach of adjusting the reference trajectory is explained. For a motion control system with two or three axes, there are four parts in the control framework, as shown in Fig. 4. The first part represents two-axis or three-axis plant and inner controllers of individual axes that guarantee closed-loop stability. This part receives G-code generated by digital curves in G01 lines. The inner controller would not affect the design of CEE and CEC as long as the stability of P_i is guaranteed. The second part is CEE, which computes the contour error based on the desired trajectory and actual trajectory which are represented by discrete data points through sampling. The third part is error analysis where the root mean square error and maximum error are used to observe the error variation in the iterative process and verify whether the machining requirements have been met. The last part is CEC to determine the error compensation and generate a new reference trajectory. By going through these four parts, a discrete reference trajectory is obtained to generate new G-code based on G01.

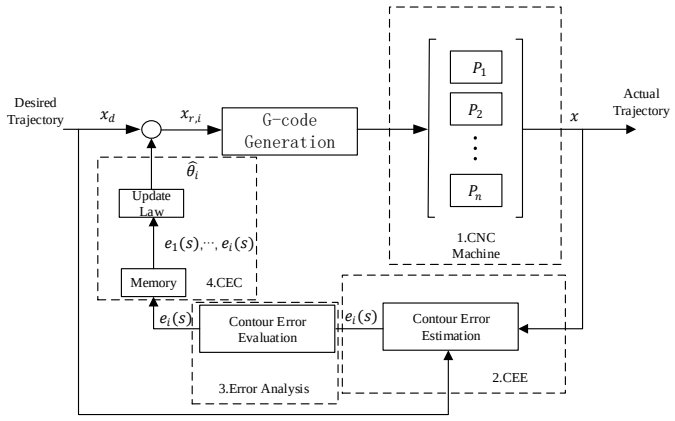


Fig. 4: The control block diagram

A. CNC system model

Fig. 5(a) shows a schematics of the dynamics model of the motion control system. In order to analyze this system, it can be simplified as a mechanical model shown in Fig. 5(b), which can be written as:

$$M\,\ddot{x}(t) + B\,\dot{x}(t) + F = u + d \tag{12}$$

where x is the position, u is the control input and d is the unknown disturbance. M, B, F are the inertia, coefficient of viscous friction and coulomb friction, respectively.

For the following ILC design, we assume that the disturbance model is

$$d = -\theta_i \tag{13}$$

where $\theta_i(s) = \theta_{i-1}(s)$ with s representing the displacement that will be explained later. This model indicates that the disturbance is repetitive in the displacement.

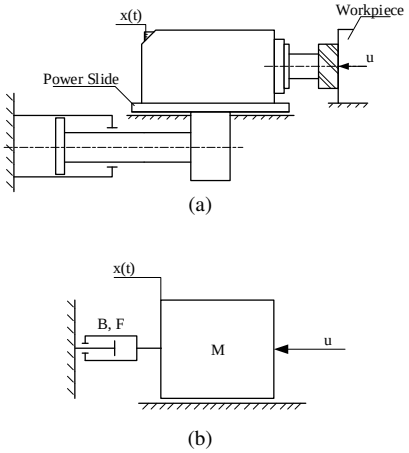


Fig. 5: A schematics of the dynamics model of the motion control system (a) and mass-damping system (b)

We design a controller including two parts, i.e., $u = u_1 + u_2$ where u_1 is used to compensate for the system dynamics, as below

$$u_1 = M\,\ddot{x}_e(t) + B\,\dot{x}_e(t) + F \tag{14}$$

$\dot{x}_e$ is an auxiliary variable defined as

$$\dot{x}_e = \dot{x}_d - Le,\ e = x - x_d \tag{15}$$

where x_d is the desired trajectory, e is tracking error and L is a positive definite matrix. u_2 is used to introduce feedback, as below

$$u_2 = -K[\dot{e} + L(x - x_r)] = -K[e_v + L(x_d - x_r)] \tag{16}$$

where $e_v = \dot{e} + Le$, x_r is the reference trajectory and K is the gain of position control loop.

By substituting (14) and (16) into (12), we obtain the error dynamics

$$M\,\dot{e}_v(t) + (B + K)e_v(t) = -KL(x_d - x_r) + d \tag{17}$$

From the above equation, we find that if the disturbance $d = 0$, choosing the reference trajectory $x_r = x_d$ will lead to $e_v = 0$ and thus $e = 0$ according to the definition of e_v. Since $d \neq 0$, we need to design x_r so that $e = 0$ is still guaranteed.

B. Transformation from time domain to spatial domain

As we assume that the disturbance d is repetitive in displacement, we consider transferring the system dynamics from time domain to spatial domain. First, let us discuss the relationship between the spatial and temporal coordinates. As have been defined before, t is the time, s is the displacement of the system along the path and $v = \frac{ds}{dt}$ is the speed of the system in this direction. Therefore, we can obtain

$$\frac{d}{dt} = \frac{d}{ds}\frac{ds}{dt} = \frac{d}{ds}v \tag{18}$$

The spatial differentiator, or the $\bigtriangledown$ operator [29], is defined below and linked to the temporal differentiator:

$$\bigtriangledown = \frac{d}{ds} \tag{19}$$

In order to facilitate the conversion between t and s, let us analyze the relationship between the temporal coordinate t and spatial coordinate s. From $ds = vdt$ we have $s = \int_0^t v(\tau)d\tau$. When the system's speed $v > 0$, s is a strictly increasing function of t, hence the relationship between t and s is bijective. The function $s = f(t)$ is analytic and the inverse function $t = f^{-1}(s)$ exists globally. Therefore as a variable, $v(t)$ can also be expressed as a spatial function $v(f^{-1}(s))$.

Based on above discussions, we can transform the error dynamics in Eq. (17) into the spatial domain, as below

$$Mv \bigtriangledown e_v(s) + (B+K)e_v(s) = -KL[x_d(s) - x_r(s)] + d \tag{20}$$

where $x_d(s)$ is the desired position and $x_r(s)$ is the reference position in space domain and $e_v(s)$ is the error in space domain, including the contour error $e(s) = x(s) - x_d(s)$.

C. Reference modification

According to (20), we can make the contour error $e(s)$ converge to 0 if $-KL(x_r - x_d) = d$. Therefore, according to (13), in the i-th iteration we design x_r as below

$$x_{r,i}(s) = x_d(s) + \frac{1}{KL}\hat{\theta}_i \tag{21}$$

which is equivalent to

$$x_{r,i}(s) = x_{r,i-1}(s) + \delta\hat{\theta}_i \tag{22}$$

where $\hat{\theta}_i$ is the estimate of θ_i and is updated iteratively as below

$$\delta\hat{\theta}_i = \hat{\theta}_i - \hat{\theta}_{i-1} = Qe_{v,i}(s) \tag{23}$$

where Q is a positive definite gain matrix.

According to (13), (20) and (21), in the i-th iteration we have

$$Mv \bigtriangledown e_{v,i} + (B+K)e_{v,i} = \tilde{\theta}_i \tag{24}$$

where $\tilde{\theta}_i = \hat{\theta}_i - \theta_i$. Furthermore, we can get

$$M \bigtriangledown e_{v,i} + \frac{1}{v}Be_{v,i} + \frac{1}{v}Ke_{v,i} = \frac{1}{v}\tilde{\theta}_i \tag{25}$$

From this equation, we can easily see that if $\tilde{\theta}_i = 0$ then $e_{v,i} \to 0$ and thus $e_i \to 0$, indicating that the contour error converges to 0. Therefore, the objective of the adaptation of $\hat{\theta}_i$ is to minimize $\tilde{\theta}_i$, which can be carried out through minimizing the cost function as below

$$J_c(s) = \frac{1}{2}\int_{s-S}^{s} \frac{1}{v}(\tilde{\theta}_i^T Q^{-1}\tilde{\theta}_i)d\tau \tag{26}$$

where S is the displacement of the system in one iteration. Correspondingly, the reduction of the contour error can be obtained by minimizing the cost function

$$J_e(s) = \frac{1}{2}e_{v,i}^{\mathrm{T}} M e_{v,i} \tag{27}$$

Consequently, we consider a combined cost function

$$J = J_c + J_e \tag{28}$$

that will be shown to decrease as the iteration number increases.

D. Convergence analysis

In particular, we consider the space derivative of J_e, as below

$$\begin{aligned}\bigtriangledown J_e(s) &= e_{v,i}^T M \bigtriangledown e_{v,i} + \frac{1}{2}e_{v,i}^T \frac{1}{v}\dot{M}e_{v,i} \\ &= e_{v,i}^T M \bigtriangledown e_{v,i} + e_{v,i}\frac{1}{v}Be_{v,i}\end{aligned} \tag{29}$$

where we use the skew-symmetry property [30], i.e.

$$e_{v,i}^T \dot{B}e_{v,i} = 2e_{v,i}^T Be_{v,i} \tag{30}$$

Considering Eq. (25), the above equation can be rewritten as

$$\begin{aligned}\bigtriangledown J_e(s) &= e_{v,i}^T[M \bigtriangledown e_{v,i} + \frac{1}{v}Be_{v,i}] \\ &= e_{v,i}^T[\frac{1}{v}\tilde{\theta}_i - \frac{1}{v}Ke_{v,i}]\end{aligned} \tag{31}$$

Let us then consider the difference between J_c of two consecutive iterations as below

$$\begin{aligned}\Delta J_c &= J_c(s) - J_c(s-S) \\ &= \frac{1}{2}\int_{s-S}^{s} \frac{1}{v}\tilde{\theta}_i^T Q^{-1}\tilde{\theta}_i - \frac{1}{v}\tilde{\theta}_i^T(\tau - S)Q^{-1}\tilde{\theta}_i(\tau - S)d\tau\end{aligned} \tag{32}$$

where we compute

$$\begin{aligned}&\frac{1}{v}\tilde{\theta}_i^T(\tau)Q^{-1}\tilde{\theta}_i(\tau) - \frac{1}{v}\tilde{\theta}_{i-1}^T(\tau - S)Q^{-1}\tilde{\theta}_{i-1}(\tau - S) \\ &= (\frac{1}{v}\tilde{\theta}_i^T(\tau)Q^{-1}\tilde{\theta}_i(\tau) - \frac{1}{v}\tilde{\theta}_i^T(\tau)Q^{-1}\tilde{\theta}_{i-1}(\tau - S)) + \\ &\quad (\frac{1}{v}\tilde{\theta}_i^T(\tau)Q^{-1}\tilde{\theta}_{i-1}(\tau - S) - \frac{1}{v}\tilde{\theta}_{i-1}^T(\tau - S)Q^{-1}\tilde{\theta}_{i-1}(\tau - S)) \\ &= -\frac{1}{v}\tilde{\theta}_i^T(\tau)Q^{-1}\delta\hat{\theta}_i - \frac{1}{v}\tilde{\theta}_{i-1}^T(\tau - S)Q^{-1}\delta\hat{\theta}_i \\ &= -(2\frac{1}{v}(\tilde{\theta}_i^T(\tau)) + \delta\hat{\theta}_i)Q^{-1}Qe_{v,i}(\tau) \\ &\le -2\frac{1}{v}\tilde{\theta}_i^T(\tau)e_{v,i}(\tau) \\ &= -2\frac{1}{v}e_{v,i}\tilde{\theta}_i(\tau)\end{aligned} \tag{33}$$

Thus, we have

$$\Delta J_c \le -\int_{s-S}^{s}[\frac{1}{v}e_{v,i}^T\tilde{\theta}_i(\tau)]d\tau \tag{34}$$

From (31), we can get the variation of J_e as follows

$$\Delta J_e = \int_{s-S}^{s} e_{v,i}^T[\frac{1}{v}\tilde{\theta}_i(\tau) - \frac{1}{v}Ke_{v,i}(\tau)]d\tau \tag{35}$$

By considering Eqs. (34) and (35), we have

$$\Delta J = \Delta J_c + \Delta J_e \le -\int_{s-S}^{s}\frac{1}{v}e_{v,i}^T(\tau)Ke_{v,i}(\tau)d\tau \tag{36}$$

Because K is positive-definite, we can get $\Delta J \le 0$.

By Ineq. (36), we have shown that the function J does not increase when the iteration number increases for $s \in [0, S]$. Then, we will have the boundedness of J if we show that J in the first iteration is bounded, i.e., $J_{i=1} < \infty$.

Considering the spatial derivative of $J_{i=1}$ as below

$$\bigtriangledown J_{i=1} = \bigtriangledown J_c + \bigtriangledown J_e \tag{37}$$

According to Ineq. (36), we have

$$\bigtriangledown J_{i=1} \leq -\frac{1}{v} e_{v,i} K e_{v,i} \leq 0 \tag{38}$$

Therefore, by integrating $\bigtriangledown J_{i=1}$ from 0 to s, we obtain

$$J_{i=1} - J_{i=1}(0) \leq 0 \tag{39}$$

Since $e_{v,1}(0)$ and M are bounded, $J_e(0)$ is bounded according to its definition. Since the displacement S and true values of parameters θ are bounded and $\hat{\theta}$ is initialized as zero before the first iteration, $J_c(0)$ is bounded. Therefore, $J_{i=1}(0)$ is bounded, and thus $J_{i=1}$ is bounded.

Finally, by Ineq. (36), we have

$$\Delta J \leq -\int_{s-S}^{s} \frac{1}{v} e_{v,i} K e_{v,i} d\tau \tag{40}$$

From the above inequality, we obtain

$$J - J_{i=1} \leq -\sum_{j=1}^{i-1} \int_{s-S}^{s} \frac{1}{v} e_{v,i} K e_{v,i} d\tau \tag{41}$$

which leads to

$$J_{i=1} \geq -\sum_{j=1}^{i-1} \int_{s-S}^{s} \frac{1}{v} e_{v,i} K e_{v,i} d\tau \tag{42}$$

Since $J_{i=1}$ is bounded, we can conclude that $\|e_{v,i}\| \to 0$ when the iteration number $i \to \infty$.

IV. Experiments

A. Experimental setup

The proposed method of contour control is tested on our NC machine shown in Fig. 6, which uses a Googol numerical control system with actuators of Sanyo. The rotary encoders serve as position sensors with a resolution of 32768p/r, and the pitch of motor is 5mm. The controller board executes algorithms at a sampling frequency of 500Hz. Note that a different sampling time or resolution will lead to different data points, and thus will affect the error, but the proposed method achieves better performance than other methods, regardless of the sampling time. In the experiment, the NC machine receives the G code based on G01, and it collects data at the same sampling time. The displacement s is the distance traveled so far, i.e. increment of displacement is related to the sampling time and is the distance between two adjacent points.

The following two performance indexes are employed to evaluate the quality of the contour error control strategy [31]:

1) $e_{rms}(s) = \sqrt{\frac{1}{S}\int_0^S |e(s)|^2 ds}$ is the root-mean-square (RMS) value of the contour error $e(s)$ defined in the proposed method and S represents the total length of a trajectory.

2) $e_{max}(s) = max|e(s)|$ is the maximum absolute (MAX) value of the contour error over the whole length of the trajectory.

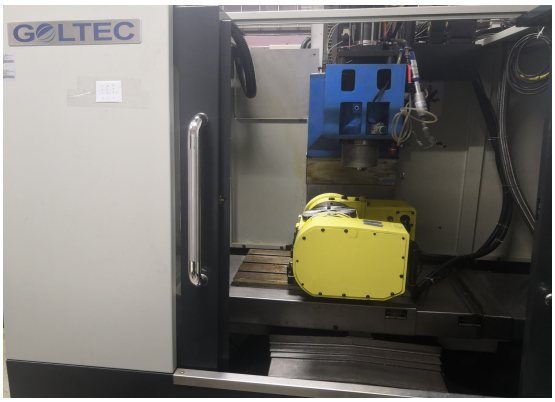


Fig. 6: Multi-axis CNC machine

B. Experimental results

The experiments are conducted to verify the CEE and CEC methods based on modification of the reference trajectory in the spatial domain. For comparison, the method of modifying the reference trajectory based on the tracking error in the time domain is also tested. In addition, how the reference trajectory affects the contour error is investigated, and the influence of learning rate on experimental results is analyzed. In the experiment, we give limits on maximum velocity and maximum acceleration in G code, with the maximum feed rate $1000mm/min$ and the maximum feed acceleration $1000mm/min^2$.

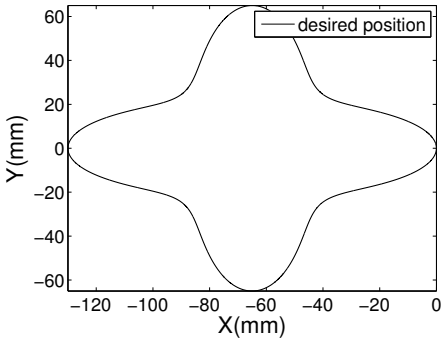


Fig. 7: A fourth-order ellipse trajectory

1) Modification of reference trajectory based on tracking error: Before verifying the method proposed in this paper, first we use the tracking error, i.e. the error in time domain, to modify the reference trajectory and study its performance. A fourth-order ellipse is used as the reference trajectory (see Fig. 7), which is defined as

$$\begin{cases} x(t) = \frac{Pcost}{1-kcos(4t)} \\ y(t) = \frac{Psint}{1-kcos(4t)} \\ P = A(1-k) \end{cases} \tag{43}$$

where $A = 65$ and $k = 0.3$, and the sampling interval of parameter t is 5ms. Likewise, a different interval will lead to different data points, and thus will affect the error, but the better performance is achieved by the proposed method, regardless of the interval. The trajectory is discretized with a sampling time to obtain a digital curve and generate the G code of the original trajectory based on G01.

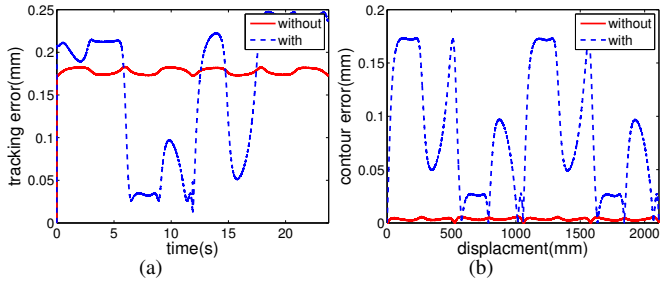


Fig. 8: Tracking error (a) and contour error (b) with and without modification of reference trajectory based on tracking error

By computing the tracking error $e(t) = x(t) - x_d(t)$, we modify the reference trajectory with a learning rate of $Q = 0.8$. The tracking error is given in Fig. 8 (a) and the contour error is shown in Fig. 8 (b). From these figures, it is found that the tracking error varies irregularly in different regions, and the contour error significantly increases from 6μm to 170μm. Therefore, we can draw a conclusion that the method based on the tracking error is not helpful in improving the contour accuracy when the tool speed changes in each iteration.

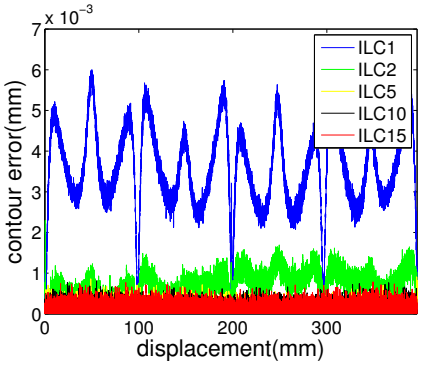


Fig. 9: Contour errors of the fourth-order ellipse with sILC

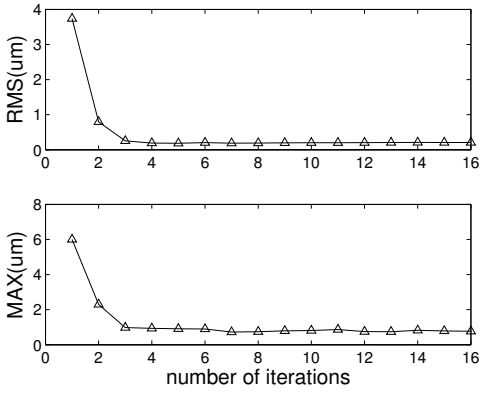


Fig. 10: RMS and MAX errors of the fourth-order ellipse

2) Fourth-order ellipse trajectory: The same fourth-order ellipse trajectory is considered in this subsection, while the proposed CEC based on the contour error is tested. The variation of the contour error during the iteration process is shown in Fig. 9, and it can be found that the contour error converges to a small value after 5 iterations. In the first iteration (without reference modification), $e_{max}(s)$ is 6.0002μm and $e_{rms}(s)$ is 3.7387μm. And the change of $e_{rms}(s)$ and $e_{max}(s)$ with respect to the iterations of the contour control scheme is shown in Fig. 10. From this figure, we can find that $e_{rms}(s)$ and $e_{max}(s)$ decrease with respect to the iterations, and eventually converge to 0.205μm and 0.75μm, respectively. Compared to the results in the first iteration, $e_{max}(s)$ is reduced by 87% and $e_{rms}(s)$ reduced by 94%, which illustrate the effectiveness of the proposed method.

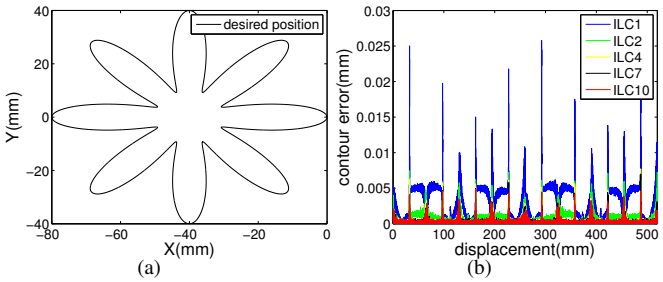


Fig. 11: A flower curve (a) and contour error changes with sILC (b)

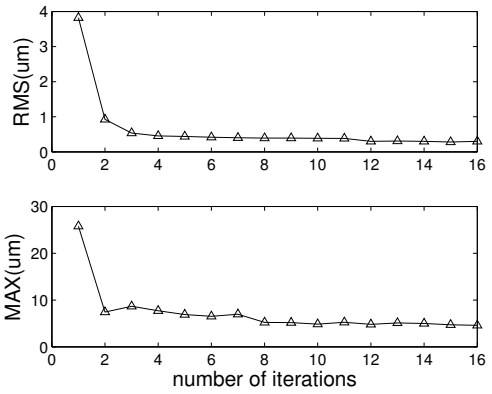


Fig. 12: RMS and MAX errors of the flower curve

3) Flower curve: In order to test the proposed method for large curvature trajectories, we use a flower curve given by

$$\begin{cases} r(t) = a + b\cos(ct) \\ x(t) = r(t)\cos(t) \\ y(t) = r(t)\sin(t) \end{cases} \tag{44}$$

where $a = 25$, $b = 15$ and $c = 5$, and the sampling interval of parameter t is also 5ms. This trajectory is shown in Fig. 11 (a), with the same method to obtain the digital curve and further obtain the G code.

Like in the previous experiment, we implement the proposed method of contour control and obtain the changes of contour error throughout the iteration process in Fig. 11 (b). From this figure, we can find that the contour error in low-curvature areas is about 5μm and in high-curvature areas is about 25μm in the first iteration. This is not surprising as due to the characteristics of CNC machine tools and speed planning, the contour error for high-curvature trajectories is usually larger than the low-curvature ones. Then, with the progress

of iterations, it can be found that the contour error decreases gradually and significantly.

The RMS and MAX values of contour error are shown in Fig. 12. From this figure, we can find that $e_{rms}(s)$ decreases form 3.8222μm to 0.290μm after iterations, which is very close to that of the fourth-order ellipse in Fig. 10, indicating that the large contour errors at high-curvature areas have little effect on $e_{rms}(s)$ because those large contour errors are local. Conversely, they have a significant effect on $e_{max}(s)$, which is larger than that of the ellipse in Fig. 10. Despite this, $e_{max}(s)$ decreases from 25μm to 5μm when the iteration number increases to 8. Compared to the results in the first iteration, $e_{rms}(s)$ is reduced by 93.8% and $e_{max}(s)$ reduced by 80%. These results show that the proposed method is also effective for large-curvature trajectories.

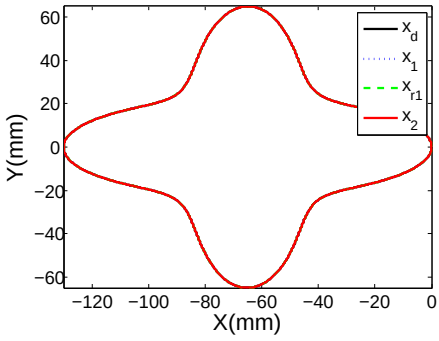


Fig. 13: Trajectory modification of the fourth-order ellipse

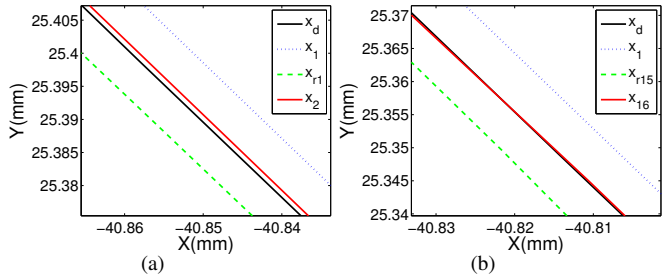


Fig. 14: The first iteration (a) and the last iteration (b) trajectory modification of the fourth-order ellipse

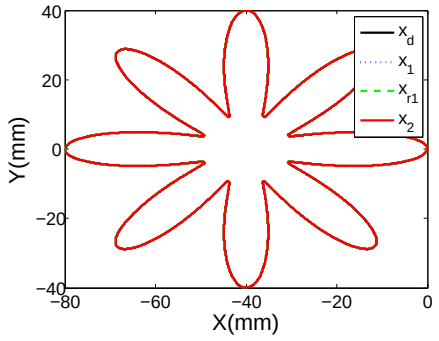


Fig. 15: Trajectory modification of the flower curve

4) Modifications of reference trajectory: Next, we analyze what happens to the reference trajectory when it is modified

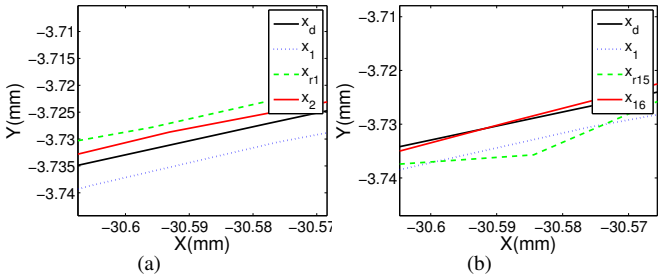


Fig. 16: The first iteration (a) and the last iteration (b) trajectory modification of the flower curve

to improve the contour tracking accuracy. For the fourth-order ellipse, with the learning rate 0.8, the trajectory modification is shown in Fig. 13, where x_d, x_r and x respectively represent the desired trajectory, reference trajectory and actual trajectory. Figs. 14 (a) and (b) present the adjustments of the fourth-order ellipse for the first and last iterations, respectively. From these figures, we can find that as the actual trajectory is on the right hand side of the desired trajectory, the reference trajectory compensates for their difference by setting itself on the left hand side of the desired one so the actual trajectory gets closer to the desired trajectory. A similar result can be found for the flower curve in Fig. 15, with Figs. 16 (a) and (b) presenting the adjustments of the flower curve for the first and last iterations, respectively. A different result is that after several iterations, the reference trajectory becomes less smooth, although the contour error does not diverge. This is more obvious on a trajectory with a large curvature. Since the contour error can converge to a small value in the first few iterations, when the error meets the requirements we can terminate the modification of the reference trajectory which has been found to be able to effectively solve this problem.

5) Effects of learning rate: A constant learning rate of $Q = 0.8$ was used in previous experiments and ensured good results. In this subsection, the experimental results are studied by increasing and decreasing the learning rate.

The results for $Q = 0.3$ are presented in Fig. 17, where we find that $e_{rms}(s)$ and $e_{max}(s)$ decrease more slowly compared with that of $Q = 0.8$. Nevertheless, the converging values are similar, with ε_{rms} decreased from 3.7387μm to 0.1846μm and ε_{max} from 6.002μm to 0.6410μm, respectively.

The results for $Q = 1.5$ are presented in Fig. 18. Surprisingly, the decrease of ε_{rms} and ε_{max} is also slower than that of $Q = 0.8$. This may be due to overshoot of the contour error during the modification of the reference trajectory. $e_{rms}(s)$ decreases from 3.7387μm to 0.2686μm and $e_{max}(s)$ from 6.002μm to 0.9820μm, which suggest a converging performance similar to that of $Q = 0.8$.

Finally, when we increase the learning rate to 2.2, we can find that $e_{rms}(s)$ and $e_{max}(s)$ diverge as shown in Fig. 19. The reason is that when the learning rate is too large, the contour error overshoots to the opposite direction to the desired one, and its absolute value exceeds the original contour error. In the subsequent iterations, the same process continues so it diverges. Therefore, a good trade-off of fast convergence and

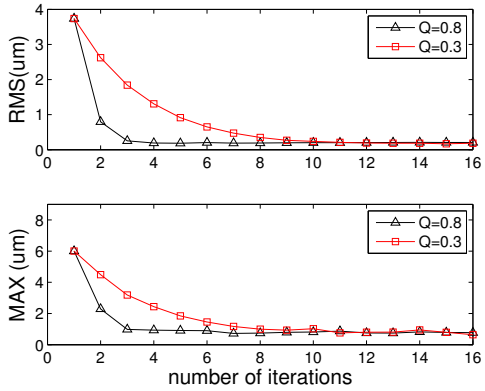


Fig. 17: RMS and MAX errors of fourth-order ellipse for $Q = 0.3$

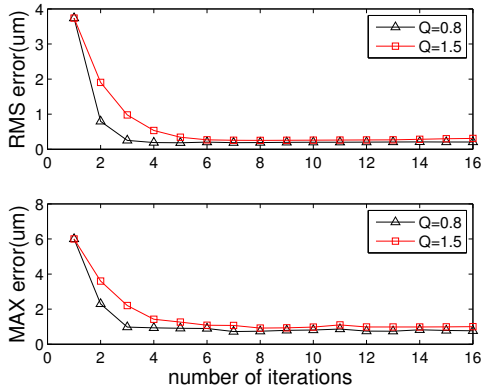


Fig. 18: RMS and MAX errors of fourth-order ellipse for $Q = 1.5$

robustness needs to be considered when setting the learning rate.

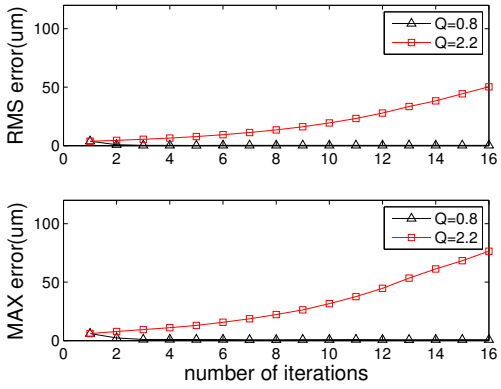


Fig. 19: RMS and MAX errors of fourth-order ellipse for $Q = 2.2$

V. Conclusion

This paper presented a method to improve contour tracking accuracy in numerical control machining. For digital curves, we used Hermite-spline to obtain a smooth free curve and developed a contour error estimation method. The reference trajectory was modified based on the estimated contour error in spatial domain. The stability and convergence of the proposed method were proved in theory. These results were supported with experimental results on a two-axis CNC system.

From the experimental results, we pointed out that using tracking error in time domain to modify the reference trajectory could not reduce the contour error, while using the estimated contour error in spatial domain achieved significant reduction of contour error for both smooth and high-curvature trajectories. The influence of reference trajectory change and the effects of learning rate on contour tracking were investigated.

This method of modifying the reference trajectory does not interfere with the low-level controller, so it is applicable to many commercial CNC systems and is relatively easy to implement. Our future works will focus on dealing with high-curvature trajectories using the same framework but possibly with varying learning rates.

The proposed method can be used for a system of up to 3 axes in Cartesian space but not readily for a 5-axis system. While extension to 5-axis systems is in our plan, we notice that there are still many applications requiring 2-axis and 3-axis systems, e.g. laser cutting and positioning tables. Moreover, the proposed iterative learning is useful for mass production, e.g. mobile phone screens and cases, anti-lock brakes for cars, alloy wheels, etc. Finally, the position sensors used in this work are encoders, which do not consider the effect of the mechanical system. For a more accurate measurement, optical gratings may be used.

References

[1] M. F. Corapsiz and K. Erenturk, "Trajectory tracking control and contouring performance of three-dimensional CNC," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 4, pp. 2212–2220, 2016.

[2] S. Oh and K. Kong, "Two-degree-of-freedom control of a two-link manipulator in the rotating coordinate system," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 9, pp. 5598–5607, 2015.

[3] H.-S. Park, P. H. Chang, and D. Y. Lee, "Concurrent design of continuous zero phase error tracking controller and sinusoidal trajectory for improved tracking control," *Journal of Dynamic Systems, Measurement, and Control*, vol. 123, no. 1, pp. 127–129, 2001.

[4] G.-C. Chiu and M. Tomizuka, "Contouring control of machine tool feed drive systems: a task coordinate frame approach," *IEEE Transactions on Control Systems Technology*, vol. 9, no. 1, pp. 130–139, 2001.

[5] C. Hu, B. Yao, and Q. Wang, "Coordinated adaptive robust contouring control of an industrial biaxial precision gantry with cogging force compensations," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 5, pp. 1746–1754, 2010.

[6] Y. Koren, "Cross-coupled biaxial computer control for manufacturing systems," *Journal of Dynamic Systems, Measurement, and Control*, vol. 102, no. 4, pp. 265–272, 1980.

[7] Y. Koren and C.-C. Lo, "Variable-gain cross-coupling controller for contouring," *CIRP annals*, vol. 40, no. 1, pp. 371–374, 1991.

[8] F. Huo and A.-N. Poo, "Improving contouring accuracy by using generalized cross-coupled control," *International Journal of Machine Tools and Manufacture*, vol. 63, pp. 49–57, 2012.

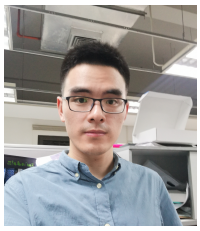
[9] L. Tang and R. G. Landers, "Predictive contour control with adaptive feed rate," *IEEE/ASME Transactions On Mechatronics*, vol. 17, no. 4, pp. 669–679, 2011.

[10] S.-S. Yeh and P.-L. Hsu, "Estimation of the contouring error vector for the cross-coupled control design," *IEEE/ASME transactions on mechatronics*, vol. 7, no. 1, pp. 44–51, 2002.

[11] R. Shi, Y. Lou, Y. Shao, J. Li, and H. Chen, "A novel contouring error estimation for position-loop cross-coupled control of biaxial servo systems," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2197–2202, IEEE, 2016.

[12] F. Huo, X.-C. Xi, and A.-N. Poo, "Generalized taylor series expansion for free-form two-dimensional contour error compensation," *International Journal of Machine Tools and Manufacture*, vol. 53, no. 1, pp. 91–99, 2012.
[13] M. R. Khoshdarregi, S. Tappe, and Y. Altintas, "Integrated five-axis trajectory shaping and contour error compensation for high-speed cnc machine tools," *IEEE/ASME Transactions on Mechatronics*, vol. 19, no. 6, pp. 1859–1871, 2014.
[14] A. Ghaffari and A. G. Ulsoy, "Dynamic contour error estimation and feedback modification for high-precision contouring," *IEEE/ASME Transactions on Mechatronics*, vol. 21, no. 3, pp. 1732–1741, 2016.
[15] X. Yang, R. Seethaler, C. Zhan, D. Lu, and W. Zhao, "A novel contouring error estimation method for contouring control," *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 4, pp. 1902–1907, 2019.
[16] Z. Wang, C. Hu, and Y. Zhu, "Dynamical model based contouring error position-loop feedforward control for multi-axis motion systems," *IEEE Transactions on Industrial Informatics*, 2019.
[17] M. Hanifzadegan and R. Nagamune, "Contouring control of cnc machine tools based on linear parameter-varying controllers," *IEEE/ASME Transactions on Mechatronics*, vol. 21, no. 5, pp. 2522–2530, 2016.
[18] Z. Kuang, G. Sun, and H. Gao, "Simplified newton-based cee and discrete-time fractional-order sliding-mode cec," *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 1, pp. 175–185, 2019.
[19] M. Yuan, Z. Chen, B. Yao, and X. Zhu, "Time optimal contouring control of industrial biaxial gantry: A highly efficient analytical solution of trajectory planning," *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 1, pp. 247–257, 2017.
[20] M. Yuan, Z. Chen, B. Yao, and J. Hu, "An improved online trajectory planner with stability-guaranteed critical test curve algorithm for generalized parametric constraints," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 5, pp. 2459–2469, 2018.
[21] Z. Jiang, J. Ding, Z. Song, L. Du, and W. Wang, "Modeling and simulation of surface morphology abnormality of 'S' test piece machined by five-axis cnc machine tool," *The International Journal of Advanced Manufacturing Technology*, vol. 85, no. 9-12, pp. 2745–2759, 2016.
[22] K. L. Barton and A. G. Alleyne, "A cross-coupled iterative learning control design for precision motion control," *IEEE Transactions on Control Systems Technology*, vol. 16, no. 6, pp. 1218–1231, 2008.
[23] M.-S. Tsai, C.-L. Yen, and H.-T. Yau, "Integration of an empirical mode decomposition algorithm with iterative learning control for high-precision machining," *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 3, pp. 878–886, 2012.
[24] K. L. Moore, M. Ghosh, and Y. Q. Chen, "Spatial-based iterative learning control for motion control applications," *Meccanica*, vol. 42, no. 2, pp. 167–175, 2007.
[25] D. J. Hoelzle and K. L. Barton, "A new spatial iterative learning control approach for improved micro-additive manufacturing," in *2014 American Control Conference*, pp. 1805–1810, IEEE, 2014.
[26] D. J. Hoelzle and K. L. Barton, "On spatial iterative learning control via 2-d convolution: Stability analysis and computational efficiency," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 4, pp. 1504–1512, 2015.
[27] J. Liu, X. Dong, D. Huang, and M. Yu, "Composite energy function-based spatial iterative learning control in motion systems," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 5, pp. 1834–1841, 2018.
[28] G. D. Knott, "Interpolating cubic splines," *Publications of the American Statistical Association*, vol. 97, no. 457, pp. 366–366, 1999.
[29] J.-X. Xu and D. Huang, "Spatial periodic adaptive control for rotary machine systems," *IEEE Transactions on Automatic Control*, vol. 53, no. 10, pp. 2402–2408, 2008.
[30] C. C. de Wit, B. Siciliano, and G. Bastin, *Theory of robot control*. Springer Science & Business Media, 2012.
[31] Z. Wang, C. Hu, Y. Zhu, S. He, M. Zhang, and H. Mu, "Newton-ilc contouring error estimation and coordinated motion control for precision multiaxis systems with comparative experiments," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 2, pp. 1470–1480, 2017.

Jiangang Li (M'09) received the BEng, MEng, PhD degrees from the Xi'an Jiaotong University, China, in 1999, 2002 and 2005, respectively. From 2007 to present, he has been an Associate Professor in control science and engineering with the School of Mechanical Engineering and Automation, Harbin Institute of Technology Shenzhen, China. From 2015 to 2016, he has been a Visiting Associate in computing and mathematical sciences with the California Institute of Technology. His general research interests include high speed and high performance control system design, motion control and motion planning.



Yiming Wang received the BEng degree from the Chang'an University, Xi'an, China, in 2017 and the MEng degree from the Harbin Institute of Technology Shenzhen, China, in 2019. He is currently a software engineer with the 28th Research Institute of China Electronics Technology Group Corporation. His research interests include motion control and high-performance control system design.

Yanan Li (S'10-M'14) received the BEng and MEng degrees from the Harbin Institute of Technology, China, in 2006 and 2008, respectively, and the PhD degree from the National University of Singapore, in 2013. Currently he is a Lecturer in Control Engineering with the Department of Engineering and Design, University of Sussex, UK. From 2015 to 2017, he has been a Research Associate with the Department of Bioengineering, Imperial College London, UK. From 2013 to 2015, he has been a Research Scientist with the Institute for Infocomm Research (I2R), Agency for Science, Technology and Research (A*STAR), Singapore. His general research interests include human-robot interaction, robot control and control theory and applications.

Wenshu Luo received the BEng degree from the Harbin Engineering University, Harbin, China, in 2016, and the MEng degree from the Harbin Institute of Technology Shenzhen, China. Her research interests include motion control and NC machining.